Multi-objective Reinforcement Learning Path Planning Model And Cognitive Load Assessment In The Context Of Music Education

Xiaoxuan Zhang*

College of Education, Cangzhou Normal University, Cangzhou, 061000, China

**Corresponding author. E-mail: m18603177889_1@163.com*

Received: Mar. 26, 2026; Accepted: Apr. 28, 2026

Assessment of Cognitive Load (CL) is important in music education but current approaches do not take into account the performance of students. This work proposes SN-ARi-RNN for CL prediction based on performance-aware personalization. The CL is pre-processed; features are extracted and selected with the help of SM-SSO algorithm along with EEG band mapping and MF analysis. SN-ARi-RNN predicts CL, whereas the student performance models states and clusters. Eventually, MOLERL conduct adaptive path planning with considering CL, group and task, achieving its convergence in 6412 ms.

Keywords: Music Education, Cognitive Load (CL), student's performance, ElectroEncephaloGram (EEG), Mental Fatigue (MF) score, Multi-Objective Reinforcement Learning (MORL).

http://dx.doi.org/10.6180/jase.202609_32.066

1. Introduction

CL impacts learning, representing mental effort affecting performance; exceeding memory limits reduces learning efficiency [1–5]. CL theory encompasses the load architecture. Traditional self-reports are subjective, neurophysiological approaches [6]. The proposed LSTM multimodal framework captures the temporal dynamic for adaptive music learning [7]. ML techniques such as SVM and DT evaluate CL from EEG features [8]. Pei, [9] investigated a multi-objective optimization method to design art education curriculum. Fu et al., [10] studied the multi-objective optimization in virtual reality task environment. Previous methods ignored performing, MF; SN-ARi-RNN enhances adaptive path planning. (Multi-Objective Kapur's Entropy Reinforcement Learning (MOKERL) balances exploration and performance. DBST-BCCAN is utilized for grouping the students based on age, gender, and class level. Density-Based Spatial Trid-Bray-Curtis Clustering with Adaptive Neighbors is a clustering method. Spiral Map-Seahorse Swarm Optimization is a combined feature selection technique. The organization of this paper is as follows: Section 2 introduces materials and methods to the proposed framework, section 3 introduces results and discussion, and section 4 concludes the work and future work.

2. Materials and methods

LSTM predicts CL from EEG, CNN-ViT fuses fMRI EEG, TAM-CLT analyzes CL, GeoGebra reduces CL. Fig. 1 illustrates the use of SM-SSO, SN-ARi-RNN, DBSTBCCAN, and MOKERL for personalized adaptive learning decisions.

Framework analyzes cognitive load (CL) using EEG, fNIRS, and eye-tracking data. fNIRS is a non-invasive technique used to measure brain activity through light signals. Then, $\hat{a}$ is preprocessed.

- **Missing Value Imputation:** The missing values in $\hat{a}$ are imputed using the mean of the non-missing values to manage incomplete information.

$$M = \frac{\sum \hat{a}}{x} \tag{1}$$

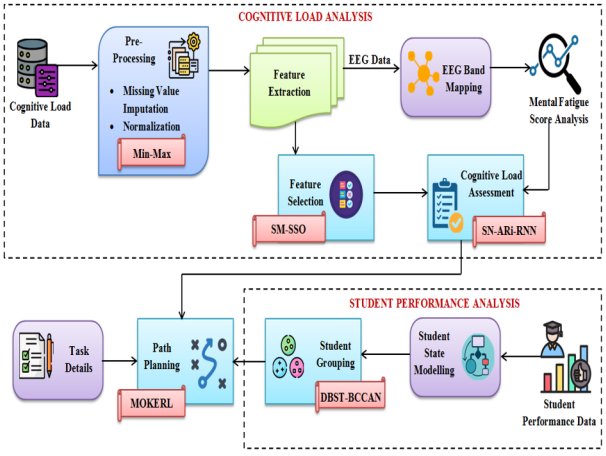


Fig. 1. Block diagram of the proposed framework

- **Min-max Normalization:** Then, minmax normalization is employed for scaling (M)to a specific range based on the maximum and minimum values.

$$D = \frac{M - M_{\min}}{M_{\max} - M_{\min}} \tag{2}$$

The features such as pupil dilation, cognitive features, and blink rate. Physiology and eyetracking measuring cognitive load; 3D mapping of EEG, fMRI and PET.

$$\aleph = \frac{1}{abc} \sum_{i=0}^{a-1} \left| \sum_{n=0}^{b-1} \breve{w}(n) y(iD) e^{jn\omega|2} \right| \tag{3}$$

$$\mathfrak{J} = \frac{\aleph(\Phi) + \aleph(\alpha)}{\aleph(\beta)} \tag{4}$$

In this section, SM-SSO chooses the most representative features. SM-SSO: Step 1 initialize agents; Step 2 evaluate fitness SN-ARi-RNN; Step 3 select best/worst; Step 4 movability exploration; Step 5 predation exploitation; Step 6 breeding diversity; Step 7 update population; Step 8 iterate until stop;

Step 9 select optimal feature subset improving model performance. Primarily, the population members in the search space are initialized. The initial population was generated within the specified bounds, sea horse in the $\left(q^{\text{th}}\right)$ dimensions is represented as,

$$\Psi_m^q = \hbar \times (\hat{u} - \bar{l}) + \bar{l} \tag{5}$$

- Movability Phase: Here, if $(\hbar = 0)$, members move spirally using SM to balance exploration and exploitation. Spiral movement allows to explore in a wide range of directions, so it keeps diversity of search, escapes local optima, and performs length search of feature subsets.

$$V = \Psi_m^q \cdot \cos(\theta) + r' \cdot \sin(\theta) \tag{6}$$

$$\begin{aligned} \bar{\Psi}_m^q = V \otimes \Big| & \Psi_m^q + \mathrm{LevV}(\ddot{\lambda}) \left(\Psi_{\text{eliec}}(\dddot{\zeta}) - \Psi_m^q\right) \\ & \bullet \dot{p} \times \dot{q} \times \dot{r} + \Psi_{\text{elie}}(\ddot{\zeta}) \Big| \end{aligned} \tag{7}$$

- Predation phase: Predation phase uses control parameters to balance exploration and exploitation, improving search efficiency. Here, if $(\hbar > 0.1)$, the members hunt the zooplankton and small crustaceans, and the position is updated as,

$$\hat{\Psi}_m^q = \begin{cases} \breve{v} * \left(\Psi_{\text{elite}}(\dddot{\zeta}) - \hbar \times \bar{\Psi}_m^q \right. \\ \left. \quad +(1 - \widetilde{v}) \otimes \Psi_{\text{elite}}(\dddot{\zeta})\right), & \text{if } \hbar > 0.1, \\ \Psi_m^q, & \text{else} \end{cases} \tag{8}$$

- Breeding Phase: Breeding phase ensures diversity, avoids convergence, improves feature selection solutions. The members with higher fitness value are considered as fathers $(\dot{\psi})$, and the members with the lowest scores are considered as mothers $(\ddot{\psi})$.

$$\dot{\psi} = \dddot{\zeta}_{\text{high}} \left(\frac{2}{X}\right) \tag{9}$$

$$\ddot{\psi} = \dddot{\zeta}_{\text{low}} \left(\frac{X(X+2)}{2}\right) \tag{10}$$

$$\psi_{\text{off}} = \hbar(\dot{\psi}) + (1 - \hbar)\ddot{\psi} \tag{11}$$

Let $(\mathfrak{J})$, and (G) are denoted by $(\hat{\mathfrak{J}})$. Spectral normalization is efficient at stabilizing training by controlling gradients, avoiding overfitting and enhancing generalization. The Adaptive Rich (ARi) (λ) activation function is utilized. SN-ARi-RNN evaluates CL via spectral normalization to achieve stable training and ARi activation to avoid dead neurons. The structure of SN-ARiRNN with input, hidden recurrent layers, and output layer for CL classification is depicted in Fig. 2.

The weight & bias of network are regularized by SN regularization to stabilize the training,

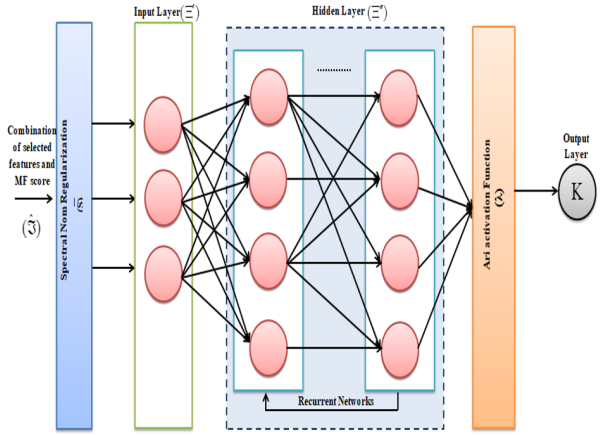


Fig. 2. Architecture of SN-ARi-RNN classifier

Algorithm 1. Pseudo-code for SN-ARi-RNN

Require: Input data $\hat{\mathfrak{J}}$
Ensure: Cognitive load level K
Initialize δ'', $\dot{\xi}$, $\bar{f}$, and Q_{t-1}
for each input sample $\hat{\mathfrak{J}}$ **do**
  Regularize the current input ▷ SN regularization
  Compute:
$$\bar{s} = \Gamma + \delta'' \times \sum \|\ddot{W}\|$$
  Implement ARi activation
  Compute:
$$\lambda = \hat{\mathfrak{J}} \cdot \frac{1}{\left(1 + e^{-\beta' \hat{\mathfrak{J}}}\right)^{\bar{\alpha}}}$$
  Evaluate the input layer
  Compute:
$$\Xi' = \lambda \left(\dot{\xi}\hat{\mathfrak{J}} + \vec{\xi} Q_{t-1}\right) + \bar{f}$$
  Determine hidden layer Ξ''
  Estimate output layer Ξ'''
  Activate output layer
  Compute:
$$K = \text{softmax}\left(\bar{\xi}\Xi'''\right) + f^{\infty}$$
  Obtain cognitive load level K
  if determined cognitive load is satisfied **then**
    Terminate
  else
    Continue iteration
return K

$$\bar{s} = \Gamma + \delta'' * \sum \left\|\ddot{W}\left(\delta''\right)\right\| \tag{12}$$

- Input layer: Here, the input layer (Ξ') prepares the current input $(\hat{\mathfrak{J}})$ at time (t) and the previous hidden state (Q_{t-1}). inputs are activated by Ari activation(λ). Ari activation (λ) as,

$$\lambda = \hat{\mathfrak{J}} \cdot \frac{1}{\left(1 + e^{-\beta' \cdot \hat{\mathfrak{J}}}\right)^{\bar{\alpha}}} \tag{13}$$

$$\Xi' = \lambda \left(\dot{\xi} \times \hat{\mathfrak{J}} + \vec{\xi} \times Q_{t-1}\right) + \bar{f} \tag{14}$$

- Hidden Layer: Information from previous time steps is captured in the hidden layer (Ξ'')

$$\Xi'' = \lambda \left(\ddot{\xi} * \hat{\mathfrak{J}} + \check{\xi} * Q_{t-1}\right) + f^{\circ} \tag{15}$$

- Output layer: Next, (Ξ'') computes the output at time (t) in the output layer (Ξ'''),

$$\Xi''' = \left(\Xi'' \times \dddot{\xi}\right) + f^{\circ\circ} \tag{16}$$

$$K = \text{softmax}\left(\hat{\xi} \cdot \Xi'''\right) + f^{\infty} \tag{17}$$

$$K = \{K_1, K_2, K_3\} \tag{18}$$

The Trid function chooses the epsilon (ε) value based on the K-distance graph, the nearby points within the epsilon value are expanded using Bray-Curtis (BC) distance (Γ'). DBST-BCCAN improves clustering using Trid epsilon and BrayCurtis distance, handling multimodal data. Primarily, (η) from the state modeling is initialized.

$$\eta = (\eta_1, \eta_2 \ldots . \eta_d) \tag{19}$$

$$\varepsilon^{\check{T}} = \sum_{j=1}^{d} \left(\eta_j - 1\right)^2 - \sum_{j=2}^{d} \eta_j \cdot \eta_{j-1} \tag{20}$$

$$\zeta_{\min} = \eta \in \varepsilon^{\check{T}} \tag{21}$$

$$\eta_{\text{core}} = \begin{cases} \eta_{\text{core}} & \text{If } (\dot{\eta} \geq \zeta_{\min}) \\ \text{noise} & \text{else} \end{cases} \tag{22}$$

$$\Gamma' = \frac{\sum_{j=1}^{k} \left|\dot{\eta}_j - \dot{\eta}_{j+1}\right|}{\sum_{j=1}^{k} \left(\dot{\eta}_j + \dot{\eta}_{j+1}\right)} \tag{23}$$

$$\chi = \Gamma' \left[\eta_{\text{core}} \bullet \dot{\eta}\left(\varepsilon^{\check{T}}\right)\right] \tag{24}$$

The pseudo-code for DBST-BCCAN is given below.

Algorithm 2. Pseudo-code for DBST-BCCAN

Require: Student state modeled output η
Ensure: Student grouped output $\bar{\chi}$
Initialize $\varepsilon^{\check{T}}$, $\zeta_{\min}$, Γ', η, and $\dot{\eta}$
for each input η **do**
  Evaluate epsilon value using the Trid function
  Compute:

$$\varepsilon^{\check{T}} = \sum_{j=1}^{d}(\eta_j - 1)^2 - \sum_{j=2}^{d}\eta_j\eta_{j-1}$$

  Determine minimum number of points:

$$\zeta_{\min} = \eta \in \varepsilon^{\check{T}}$$

  Estimate the core point
  if $\dot{\eta} \geq \zeta_{\min}$ **then**
    η_{core}
  else
    Noise
  Apply Bray–Curtis distance measure
  Compute:

$$\Gamma' = \frac{\sum_{j=1}^{k}\left|\dot{\eta}_j - \dot{\eta}_{j+1}\right|}{\sum_{j=1}^{k}\left(\dot{\eta}_j + \dot{\eta}_{j+1}\right)}$$

  Cluster the output
  Compute:

$$\hat{\chi} = \Gamma'\left|\eta_{\text{core}} \bullet \dot{\eta}\left(\varepsilon^{\check{T}}\right)\right|$$

  Obtain grouped output $\hat{\chi}$
return $\bar{\chi}$

These tasks are denoted by $(\hat{\gamma})$. Next, based on (K), $(\hat{\chi})$, and $(\hat{\gamma})$, the path planning for the students is performed in the upcoming section. The interaction among, student state S_t, and action A_t is modeled through a policy π that selects tasks by maximizing reward $R = [R_1, R_2, R_3]$, minimizing CL while improving performance and engagement. The optimal policy π^* is formulated as:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T}\gamma^t R\left(S_t, A_t\right)\right] \tag{25}$$

Where, Y is the discount factor and ^{T} denotes the learning horizon. Finally, based on the multiple objectives (i.e., (K), $(\bar{\chi})$, and $(\hat{\gamma})$). Kapur entropy maintains the explore-exploit balance, avoids premature convergence. Then, KE assigns the action of the agent, and it is defined by,

$$\dot{S} = \frac{1}{\beta - \alpha'''}\log\left(\frac{\sum_a P(a)^{\bar{\beta}}}{\sum_{\bar{a}} P(a)^{\alpha'''}}\right) \tag{26}$$

3. Result and discussion

This section evaluates the proposed model against methods using Python implementation. SN-ARi-RNN adopts learning rate 0.001 , spectral norm 0.01 , and ARi slope 0.1, scale 1.0 for stable training and performance. The Scaffold employs a multimodal dataset containing 86435 samples in the form of EEG, fNIRS, and eye-tracking for CL assessment. Dataset comprises EEG, fNIRS, eye-tracking data; preprocessing guarantees robust CL estimation. The dataset includes Low $(28,967)$, Medium $(28,625)$, and High $(28,843)$ CL classes with nearly balanced data distribution. The comparison of convergence time is illustrated in Fig. 3.

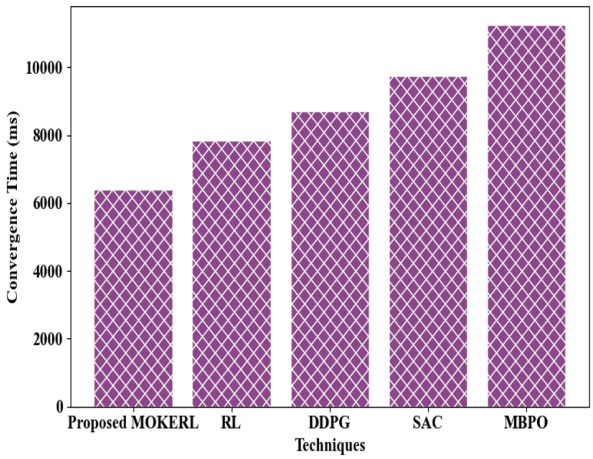


Fig. 3. Convergence Time Analysis for the proposed MOKERL

From Fig. 3 and Table 1, MOKERL attains policy entropy as 0.91 , cumulative reward as 980, the learning gain as 1.25 , and turns 6412 ms for faster convergence, which outperforms RL.

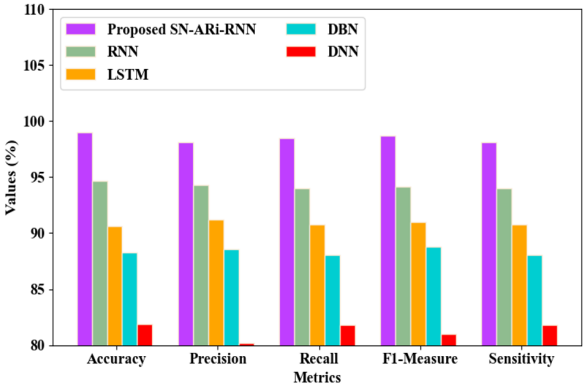


Fig. 4. Efficiency Analysis for the proposed SN-ARi-RNN

Fig. 5 and Table 3 demonstrates that DBST-BCCAN reaches 0.9456 silhouette, 0.912 ARI within 8457 ms time, outperforming DBSCAN (9452 ms, 0.8974, 0.853) under Trid epsilon and Bray-Curti's similarity.

Fig. 6 (a), (b) depicts SM-SSO attains better fitness with less time compared to SSO, BCO, ACO, BMO because of the enhanced exploration-exploitation balance. As presented in Fig. 6, SM-SSO attains 9153 ms, 35459 ms time

Table 1. Numerical Validation for the Proposed MOKERL

Technique	Convergence	Policy Time (ms)	Cumulative Entropy	Cumulative Reward Learning Gain
MOKERL	7842	0.87	873	0.92
RL	7842	0.87	873	0.92
DDPG	8698	0.79	756	0.87
SAC	9742	0.71	647	0.64
MBPO	11236	0.68	512	0.53

Table 2. Empirical Analysis for Classification

Techniques	Accuracy (%)	Precision (%)	Recall (%)	F1-measure (%)	Sensitivity (%)
SN-ARi-RNN	98.9759	98.123	98.475	98.674	98.132
RNN	94.6558	94.2657	94.0266	94.1461	94.0266
LSTM	90.6354	91.2315	90.7514	90.9914	90.7514
DBN	88.2397	88.5547	88.0695	88.8121	88.0695
DNN	81.8643	80.2106	81.8512	81.0309	81.8512

Table 3. Efficiency Evaluation for the proposed DBST-BCCAN

Techniques	Clustering Time (ms)	Silhoue tte score	ARI
Proposed DBSTBCCAN	8457	0.9456	0.912
DBSCAN	9452	0.8974	0.853
K-means	11742	0.8175	0.797
FCM	13236	0.7369	0.663
CLARA	15965	0.6589	0.512

and 89.12%, 99.21% fitness, which is better than SSO (11126 ms, 44122 ms, 87.01%, 95.67%). SVM is non temporal, CNN ignores long dependencies, LSTM unstable overfits.

As shown in Table 4, SN-ARi-RNN outperforms LSTM, CNN, OEL, and SVM, with ARi surpassing ReLU, Leaky ReLU, and PReLU. Table 5 confirms that ARi fosters both the best accuracy, and the least number of dead neurons, stable gradients, better learning, and task load classification.

4. Conclusion

The model employs SN-ARi-RNN to evaluate the CL (98.97 % accuracy), SMSSO to select the features (99.21 % fitness), DBST-BCCAN to cluster (8457 ms) and MOKERL to plan the path (980 reward), which support personalized music learning and the future work will currently under way in federated learning for multi-institution training without privacy.

5. Declarations

6. Data availability

https://www.kaggle.com/datasets/programmer3/multimodal-cognitive-load-classification-dataset

7. Conflicts of interest

The author declares that there are no conflicts of interest regarding the Table 5 confirms that ARi fosters both the best accuracy, and the least number of dead neurons, stable gradients, better learning, and task load classification. This study followed ethical standards; approval obtained before data collection.

8. Consent to participate

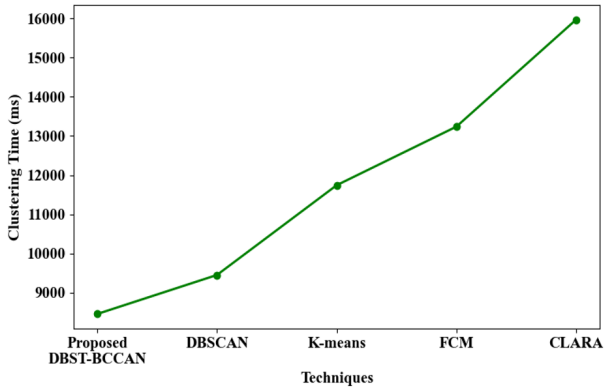
Informed consent was obtained from all participants involved in the study.

Table 4. Comparative Validation for the Proposed and Existing Methodologies

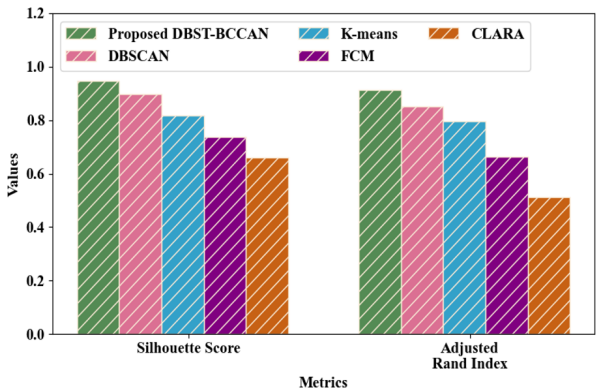
Authors Name	Methods	Accuracy (%)	Precision (%)
Proposed Work	SN-ARi-RNN	98.9759	98.123
[11]	LSTM	87.1	92
[12]	CNN	98	94
[13]	OEL	97.4	–
[14]	CNN	88.76	–
[8]	SVM	89.6	–

Table 5. Comparative Evaluation of ARi Activation Function with ReLU Variants

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	Adaptability
ReLU	94.12	93.85	93.76	Low
Leaky ReLU	95.36	95.02	95.75	Medium
PReLU	96.21	95.88	95.63	Medium
ARi (Proposed)	**98.97**	**98.65**	**98.42**	**High**

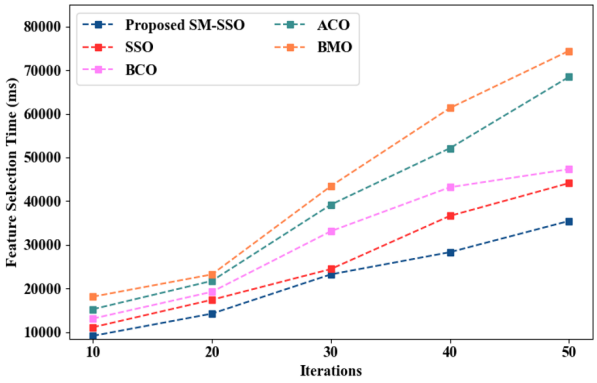


(a) Training performance analysis

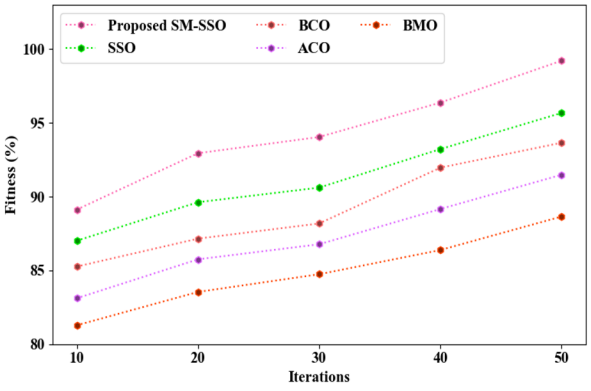


(b) Validation performance analysis

Fig. 5. Performance Analysis for the Proposed DBST-BCCAN



(a) Feature Selection Time



(b) Fitness Analysis for the proposed SM-SSO

Fig. 6. (a) Feature Selection Time and (b) Fitness Analysis for the proposed SM-SSO

9. Consent to publication

All participants provided consent for the publication of anonymized data and research findings.

10. Competing interests

The author declares no competing interests.

11. Acknowledgment

The author thanks students and faculty of Cangzhou Normal University for support and participation.

References

[1] S. Bishara, (2022) *"Linking Cognitive Load, Mindfulness, and Self-Efficacy in College Students with and without Learning Disabilities"* **European Journal of Special Needs Education** ***37***(3): 494–510. DOI: 10.1080/08856257.2021.1911521.

[2] L. Zhang, X. Wang, T. He, and Z. Han, (2022) *"A Data-Driven Optimized Mechanism for Improving Online Collaborative Learning: Taking Cognitive Load into Account"* **International Journal of Environmental Research and Public Health** ***19***(12): 6984. DOI: 10.3390/ijerph19126984.

[3] T. S. S. Rao, A. S. Tiwari, U. Mitra, and K. K. Bhagat. "Effectiveness of Marker-Based Augmented Reality Game on Computational Thinking Skills and Cognitive Load for Middle School Students". In: *2024 IEEE International Conference on Advanced Learning Technologies (ICALT)*. 2024, 222–226. DOI: 10.1109/ICALT61570.2024.00071.

[4] J. Pengelley, P. R. Whipp, and A. Malpique, (2025) *"A Testing Load: A Review of Cognitive Load in Computer and Paper-Based Learning and Assessment"* **Technology, Pedagogy and Education** ***34***(1): 1–17. DOI: 10.1080/1475939X.2024.2367517.

[5] B. Şimşek, B. Direkci, B. Koparan, M. Canbulat, M. Gülmez, and E. Nalçacıgil, (2025) *"Examining the Effect of Augmented Reality Experience Duration on Reading Comprehension and Cognitive Load"* **Education and Information Technologies** ***30***(2): 1445–1464. DOI: 10.1007/s10639-024-12864-z.

[6] X. Liu and Y. Cui, (2025) *"Eye Tracking Technology for Examining Cognitive Processes in Education: A Systematic Review"* **Computers & Education** ***229***: 105263. DOI: 10.1016/j.compedu.2025.105263.

[7] W. Chen, (2023) *"A Novel Long Short-Term Memory Network Model for Multimodal Music Emotion Analysis in Affective Computing"* **Journal of Applied Science and Engineering** ***26***(3): 367–376. DOI: 10.6180/jase.202303_26(3).0008.

[8] G. Zhu, F. Zong, H. Zhang, B. Wei, and F. Liu, (2021) *"Cognitive Load during Multitasking Can Be Accurately Assessed Based on Single Channel Electroencephalography Using Graph Methods"* **IEEE Access** ***9***: 33102–33109. DOI: 10.1109/ACCESS.2021.3058271.

[9] X. Pei, (2025) *"Research on Multi-Objective Optimization Algorithm for Optimal Design of Art Education Curriculum Based on Emotional Learning Model"* **Journal of Combinatorial Mathematics and Combinatorial Computing** ***127***: 8765–8781. DOI: 10.61091/jcmcc127a-486.

[10] Q. W. Fu, Q. H. Liu, and T. Hu, (2024) *"Multi-Objective Optimization Research on VR Task Scenario Design Based on Cognitive Load"* **Facta Universitatis, Series: Mechanical Engineering** ***22***(2): 293–313. DOI: 10.22190/FUME240122029F.

[11] G. Yoo, H. Kim, and S. Hong, (2023) *"Prediction of Cognitive Load from Electroencephalography Signals Using Long Short-Term Memory Network"* **Bioengineering** ***10***(3): 361. DOI: 10.3390/bioengineering10030361.

[12] N. Jamil and A. N. Belkacem, (2024) *"Advancing Real-Time Remote Learning: A Novel Paradigm for Cognitive Enhancement Using EEG and Eye-Tracking Analytics"* **IEEE Access** ***12***: 93116–93132. DOI: 10.1109/ACCESS.2024.3422926.

[13] J. Yedukondalu, K. Sunkara, V. Radhika, S. Kondaveeti, M. Anumothu, and Y. Murali Krishna, (2025) *"Cognitive Load Detection through EEG Lead Wise Feature Optimization and Ensemble Classification"* **Scientific Reports** ***15***(1): 842. DOI: 10.1038/s41598-024-84429-6.

[14] Y. Zhou, J. Jiang, L. Wang, S. Liang, and H. Liu, (2025) *"Enhanced Cognitive Load Detection in Air Traffic Control Operators Using EEG and a Hybrid Deep Learning Approach"* **IEEE Access** ***13***: 12127–12137. DOI: 10.1109/ACCESS.2025.3530091.